



AD FALCON API Manual

# Post-Step Actions Configuration Guide

Javad Ghorbani

March 26, 2026

# Contents

- 1 Post-Step Actions Configuration Guide 3**
- 1.1 Syntax 3
- 1.2 1 Section Placement 3
- 1.3 2 Block Structure 4
- 1.4 3 Supported Type Values 4
- 1.4.1 EstablishEquilibrium 5
- 1.5 Multiple Actions (same or different steps) 5
- 1.6 4 Element List Syntax 5
- 1.7 5 Validation Rules 5
- 1.8 Parsing Notes (robustness) 6



## 1 Post-Step Actions Configuration Guide

Post-step actions let you schedule *automatic operations* that the solver will perform immediately after a given simulation step finishes.

They are useful, for instance, for re-initialising constitutive-model state variables, enforcing equilibrium before the next stage, or any other global housekeeping task that must occur right after a step but **before** the following step begins.

### 1.1 Syntax

The % PostStepActions section contains **zero or more** @PostStepAction blocks:

```
% PostStepActions
@PostStepAction Id: <integer>
Step: <stepId>
Type: <actionType>
[ElementIDs: <list|ranges|All>]

@PostStepAction Id: <integer>
...
%%%
```

Notes:

- Each @PostStepAction ... block is independent; you can define multiple actions in the same section.
- The order of Step:, Type:, and ElementIDs: lines inside a block does not matter, but Step and Type are required.

### 1.2 1 Section Placement

Insert the block **after the referenced steps are defined** (recommended: after the final % Step Definitions block) and terminate it with %%%:

```
% PostStepActions
@PostStepAction Id: 1
Step: 2
Type: EstablishEquilibrium
ElementIDs: 101-120 205 208
%%%
```

## 1.3 2 Block Structure

Every action starts with a header line and is followed by one or more key-value rows:

```
@PostStepAction Id: <integer>
Step:           <stepId>
Type:           <actionType>
ElementIDs:     <list | range | All>      (optional)
```

**Table 1: Post-step action fields and their purpose**

Field	Purpose
<b>Id</b>	Unique integer ( $\geq 0$ ) for this action. Duplicate IDs are not allowed.
<b>Step</b>	The <b>simulation-step ID</b> after which the action will be executed. 0 means “after the initial step (step 0), before step 1 starts”.
<b>Type</b>	Kind of operation to perform (see list below).
<b>ElementIDs</b>	List parsing/validation for the action. Accepts: <i>individual IDs</i> (17 18 52), <i>ranges</i> (17-25, 100:110) and the keyword <b>ALL</b> . If omitted, the action applies to the whole mesh. (For EstablishEquilibrium, current solver behavior is global; element restriction may be extended in future.)

## 1.4 3 Supported Type Values

**Table 2: Supported post-step action types**

Type value	Description
Establish Equilibrium	Finalises equilibrium-related state at the end of the step (currently applied globally in the solver). Use this, for example, right after gravity loading to “lock in” the initial stress state before starting a coupled analysis.

*Additional action types may be added in future releases. Unsupported names will trigger a parsing error.*

Notes:

- Type values are case-insensitive (e.g. establishEquilibrium is accepted).

### 1.4.1 EstablishEquilibrium

EstablishEquilibrium is commonly used after an initial gravity/geostatic step to “lock in” the initial condition before the next step begins.

If your model uses a UMAT with custom history, FALCON calls the UMAT `initializeCustomVariable` hook again after EstablishEquilibrium so the UMAT can **condition its internal variables** (for example, set isotropic hardening size or a kinematic hardening tensor based on the current stress/void ratio/pore pressure state that has just been equilibrated).

---

## 1.5 Multiple Actions (same or different steps)

You can define multiple post-step actions by repeating `@PostStepAction` blocks. For example, schedule an equilibrium “finalize” after step 0 (before the first step starts) and after step 2:

```
% PostStepActions
@PostStepAction Id: 1
Step: 0
Type: EstablishEquilibrium

@PostStepAction Id: 2
Step: 2
Type: EstablishEquilibrium
%%%
```

If you define more than one EstablishEquilibrium action for the same Step, only one is effectively needed.

---

## 1.6 4 Element List Syntax

- **Individual IDs** 17 21 42
- **Ranges** 20-25 or 20:25 (both inclusive)
- **Mixed** 20-22 30 45:50
- **All** applies the action to every element in the mesh.

Duplicate IDs are silently ignored; unknown element numbers raise an error.

---

## 1.7 5 Validation Rules

1. **ID uniqueness** Each Id must be unique across the `% PostStepActions` section.

2. **Step existence** Step must refer to a defined simulation step (or be  $\emptyset$ ).
  3. **Mandatory fields** Id, Step, and Type are required; ElementIDs is optional.
  4. **Element checks** Every listed element must exist in the mesh; ranges expand automatically.
  5. **Multiple actions** You may define multiple @PostStepAction blocks, including multiple actions on the same Step. They are stored in input order; for EstablishEquilibrium, having more than one action on the same step is redundant (the solver checks only whether at least one exists for that step).
- 

## 1.8 Parsing Notes (robustness)

- The header token @PostStepAction is case-insensitive, and multiple leading @ characters are accepted (e.g. @@PostStepAction).
- The header line accepts common variants such as @PostStepAction Id: 1, @PostStepAction Id:1, and @PostStepAction Id 1 (case-insensitive).
- Inside a block, the keys are plain tokens (no leading @). Keys are case-insensitive (Step/Type/ElementIDs), and a trailing : is optional (e.g. step: is accepted). Other keys are rejected.
- ElementIDs accepts IDs separated by spaces, commas, or semicolons; ranges may use lo-hi or lo:hi.